

REMARKS

These remarks are made in response to the Office Action dated July 14, 2005. No amendments to the claims are made herein. Thus, claims 1-41 remain pending in the application. For the reasons set forth below, the Applicants respectfully request reconsideration and allowance of all pending claims.

Double patenting rejection

Claims 1-2, 15-16 and 25-26 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1, 3, 14 and 23 of co-pending application 10/112,920 (Hale *et al.*).

Applicants acknowledge the provisional double-patenting rejection. Should co-pending application 10/112,920 issue and the issued claims warrant a nonstatutory double-patenting rejection in the instant application, Applicants will take action at that time to overcome the double patenting rejection.

Traversal of Claim Rejections under 35 U.S.C. § 103

To establish a *prima facie* case of obviousness, there must first be some suggestion or motivation to modify a reference or to combine references, and second be a reasonable expectation of success. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. Third, the prior art reference (or references when combined) must teach or suggest all the claim limitations. M.P.E.P. § 706.02(j) from *In Re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). Where claimed subject matter has been rejected as obvious in view of a combination of prior art references, a proper analysis under § 103 requires, *inter alia*, consideration of two factors: (1) whether the prior art would have suggested to those of ordinary skill in the

art that they should make the claimed device; and (2) whether the prior art would also have revealed that in so making, those of ordinary skill would have a reasonable expectation of success. Both the suggestion and the reasonable expectation of success must be founded in the prior art, not in the Applicants' disclosure. *Amgen v. Chugai Pharmaceutical*, 927 F.2d 1200, 18 USPQ2d 1016 (Fed. Cir. 1991), *Fritsch v. Lin*, 21 USPQ2d 1731 (Bd. Pat. App. & Int'f 1991). An invention is non-obvious if the references fail not only to expressly disclose the claimed invention as a whole, but also to suggest to one of ordinary skill in the art modifications needed to meet all the claim limitations. *Litton Industrial Products, Inc. v. Solid State Systems Corp.*, 755 F.2d 158, 164, 225 USPQ 34, 38 (Fed. Cir. 1985).

The examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references. M.P.E.P. § 70602(j) from *Ex parte Clapp*, 227 USPQ 972, 973 (Bd. Pat. App. & Inter. 1985). Obviousness cannot be established by combining references without also providing evidence of the motivating force which would impel one skilled in the art to do what the patent applicant has done. M.P.E.P. § 2144 from *Ex parte Levengood*, 28 USPQ2d 1300, 1302 (Bd. Pat. App. & Inter. 1993) (emphasis added by M.P.E.P.).

Claims 1-3, 13-17, 23-27, and 33-35 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Stevens, U.S. Patent No. 6,633,976, (*Stevens*) in view of Stevens, "EFI A BIOS Vendor's Perspective" (*ReStevens*).

Claim 1 is illustrative of the independent claims, and presently recites,

1. A method comprising:

starting execution of a basic input output system (BIOS), the BIOS including a plurality of firmware modules;

42P9143

16

Examiner: Chen, Tse W.

Ser. No. 10/073,495

Art Unit: 2116

Reply to Office Communication of Nov. 16, 2005

determining ***firmware resources*** required by each of ***the plurality of firmware modules to operate***, the firmware resources required, if any, for a given firmware module provided by ***one or more other firmware modules***;

scheduling modules of the plurality of firmware modules for execution ***in consideration of the required firmware resources that are determined***; and

dispatching the scheduled modules for execution. (Emphasis Added)

In support of the rejection of claim 1 as unpatentable over *Stevens* in view of *ReStevens*, the examiner asserts the *Stevens* discloses all of the claim elements except for the element of determining resources required by the plurality of (firmware) modules. In view of this deficiency, the Examiner asserts that this element is disclosed by *ReStevens*, stating,

ReStevens discloses a method [extensible firmware interface] comprising:

Determining firmware resources [EFI drivers] requires by each of a plurality of firmware modules [EFI Option ROMs] to operate, the firmware resources required, if any, for a given firmware module provided by one or more other firmware modules [EFI PHASE 1, "Boot Process" slide; EFI drivers are firmware modules to be utilized as firmware resources by EFI Option ROMs].

Scheduling modules of the plurality of firmware modules for execution in consideration of the required firmware resources that are determined [EFI PHASE 1, "Boot Process" and "EFI PHASE 1" slides; only the EFI drivers determined to be required are loaded and scheduled for execution in order to enhance the compatibility between legacy OS and current BIOS as the scheduling of unloaded EFI drivers would render the system inoperable].

The Examiner then states,

It would have been obvious to one of ordinary skill in the art, having the teachings of *Stevens* and *ReStevens* before him at the time the invention was made, to modify the system of *Stevens* to include the well-known extensible firmware interface of *ReStevens*, in order to enhance the compatibility between legacy OS and current BIOS [ReStevens: "EFI Phase 1" slide]. One of ordinary skill in the art would have been motivated to make such a combination as it provides a way to enhance the compatibility between legacy OS and current BIOS.

As emphasized above, the plurality of modules (referred to in claim 1) for which a determination is made to what resources are required are firmware modules, and that the resources that are required are firmware resources provided by other firmware modules, and the modules are scheduled for execution in consideration of the required firmware resources that are determined. In other words, the order (schedule) in which the firmware modules are executed is dependent on the relative requirements (as provided by other modules) of the various firmware modules, such that when a given firmware module is executed, the resources required by the given module are already in place (via previous execution of one or more other modules).

As is understood by one skilled in the BIOS/Firmware art and as stated in the Background Information section of the present application,

The BIOS (also referred to herein as firmware) in a pre-memory execution environment is usually tightly bound object code that is built for a specific configuration or system design (also referred to herein as a platform). That is, different platforms typically have different BIOSs. More particularly, ***the BIOS typically includes code (also referred to herein as firmware modules)*** for providing certain functions or services, which in turn may depend on the platform. (Emphasis added)

A firmware module comprises a set of coded instructions (*i.e.*, firmware instructions). A firmware module is analogous to a software module, but is called a firmware module because it is included as part of the platform firmware (coded instructions) that are executed to prepare the platform for booting an operating system (*i.e.*, software). Additionally, in accordance with the claims herein, the resources that are determined to be required for a given firmware module are resources provided by other firmware modules.

As an illustration, the following represents a simplified example in which firmware modules are scheduled for execution based on the resource requirements of each firmware module.

| Firmware Module | Resources Required Provided by other Modules |
|-----------------|---|
|-----------------|---|

| | |
|----------|--------------|
| Module 1 | Modules 3, 4 |
| Module 2 | Module 4 |
| Module 3 | None |
| Module 4 | Module 3 |

Under the foregoing example, the scheduled order of execution would be need to be Module 3, Module 4, Module 1, Module 2 (or optionally Module 3, Module 4, Module 2, Module 1, since the ordering of Modules 1 and 2, by themselves, is irrelevant). In order to generate the schedule, the resource requirements of each firmware module are gathered and then the schedule is made such that the resource requirements will be met when each given firmware module is executed.

With respect to the Examiner's statement of "EFI drivers are firmware modules to be utilized as firmware resources by EFI Option ROMs," whether or not this may be the case is entirely unrelated to the claimed invention.

In further detail, the EFI framework enables EFI components (including drivers) to be loaded from various locations beyond the conventional locations of the platform firmware storage device (e.g., original BIOS ROM or newer flash BIOS chip) and option ROMs (i.e., ROMs on board add-on peripheral devices and cards). These new locations include EFI Option ROMs and EFI partition on a hard disk, details or which are shown in the slide entitled "What is EFI? New Partition Structure." In addition, EFI also enables EFI components to be downloaded over a network.

In "The Boot Process" slide, the illustrated EFI PHASE 1 loading merely shows EFI Option ROMs are initialized after the EFI framework is initialized (the "Initialize EFI" block). In many if not most implementations, the bulk (in some cases all) of the EFI drivers will be initialized in the "Initialize EFI" block. At this point in time, the EFI framework has yet to consider either EFI Option ROMs (should such exist) or EFI

Drivers stored on disk (should such exist). In *all* instances, the EFI Option ROMs (should such exist) will be initialized *after* the EFI framework is initialized. Accordingly, what resources (if any) that might be required by EFI Option ROMs (if any exist) have no effect on the order in which firmware modules are scheduled for execution. Furthermore, any drivers in the EFI Option ROMs haven't even been identified at the beginning of the "Initialized EFI Option ROMs" block, so it would be impossible for such unknown EFI components to have any affect on the ordering in which the other EFI components (previously executed in the "Initialize EFI" block) are scheduled to be executed.

Applicants likewise assert the following statement (also presented above) is irrelevant to the claimed invention:

Scheduling modules of the plurality of firmware modules for execution in consideration of the required firmware resources that are determined [EFI PHASE 1, "Boot Process" and "EFI PHASE 1" slides; only the EFI drivers determined to be required are loaded and scheduled for execution in order to enhance the compatibility between legacy OS and current BIOS as the scheduling of unloaded EFI drivers would render the system inoperable].

The key word here is *scheduling*. Scheduling implies generating an order in which the firmware modules are to be executed. It is further unclear how the statement, "only the EFI drivers determined to be required are loaded and scheduled for execution in order to enhance the compatibility between legacy OS and current BIOS as the scheduling of unloaded EFI drivers would render the system inoperable" has anything to do with the patentability of the claimed invention. The scheduling of execution for the firmware modules has nothing to do with compatibility with legacy OS. As before, an OS doesn't even get loaded until all of the EFI components are initialized. Furthermore, the EFI components can't, by definition, know what OS is going to be loaded, nor know whether the OS is a legacy OS or not. Thus, the order in which firmware modules are scheduled for execution is entirely independent of the operating system that is

subsequently loaded. Finally, the statement regarding scheduling of unloaded EFI drives would render the system inoperable makes no sense. Nowhere in either the claims or the current application is there any discussion of unloading firmware drivers for any purpose.

In view of the foregoing, it is clear that the combination of *Stevens* and *ReStevens* do not teach or fairly suggest all of the claim elements of amended claim 1, and thus at least the third prong of the from *In Re Vaeck* test is not met. Accordingly, independent claim 1 is clearly patentable over the cited references.

With respect to independent claim 15, this amended claim is a Beauregard claim reciting software embodied on a machine-readable medium for performing operations analogous to those recited in amended claim 1. Accordingly Claim 15 is clearly patentable over the cited references for reasons similar to those presented above in support of the patentability of claim 1.

Independent claim 25 is a system claim that presently recites,

25. A system, comprising:

- a plurality of hardware components;
- a first memory device to store a BIOS, the BIOS including a plurality of firmware modules, the BIOS further including,
 - means for determining *firmware resources* required by each of the plurality of *firmware modules* to operate, *the firmware resources required, if any, for a given firmware module provided by one or more other firmware modules*;
 - means for scheduling execution of modules of the plurality of firmware modules;
 - means for dispatching scheduled modules for execution; and
 - a processor on which the firmware modules are executed, coupled to the plurality of hardware components and the first memory device. (Emphasis added)

With respect to claim 25, the Examiners states, Stevens and ReStevens disclose each and every limitation of the claims as discussed above in reference to claims 15-17 and 23-24. Applicants respectfully assert that the element of “means for determining *firmware resources* required by each of the plurality of *firmware modules* to operate, *the firmware resources required, if any, for a given firmware module provided by one or more other firmware modules*” is clearly not disclosed, taught, or fairly suggested by either the *Stevens* reference or the *ReStevens* reference. While *Stevens* employs BIOS modules, there is no consideration of what firmware resources are required for those BIOS modules to operate. As further discussed above, the teachings of *ReStevens* also does not read on this claim element. Accordingly, amended claim 25 is clearly patentable of the cited references.

Claim 34 is a system claim that presently recites,

34. A system, comprising:

a plurality of hardware components;

a first memory device to store a BIOS, the BIOS comprising:

a plurality of firmware modules, each module of the plurality of firmware modules to provide at least one service, at least two modules providing an inter-module interface to enable each of said at least two modules to call a service provided by another module;

a core operatively coupled to the plurality of firmware modules, wherein the core, upon operation, selects for execution a set of modules from the plurality of firmware modules ***to be executed in a pre-memory execution environment prior to the initialization and availability of system memory,***

a processor coupled to the plurality of hardware components and the first memory device. (Emphasis added)

As emphasize above, Claim 34 includes the limitation that the firmware modules are ***to be executed in a pre-memory execution environment prior to the initialization and availability of system memory***, which is substantially analogous to the language recited in dependent claim 2, as well.

Applicants agree that *Stevens* discloses the use of a dispatch manager module that calls another module. However, this is clearly not done in a pre-memory execution environment prior to the initialization and availability of system memory. As shown in Fig. 6 of *Stevens*, in response to power on of a computer in step 54, the minimal initialization code 16 of the system BIOS is executed at step 55. As stated in column 5, lines 37-44,

In operation, when the computer 10a is turned on, the initialization code 16 is run to initialize the CPU 11 ***and the system memory 13***. The dispatch manager 17 is then ***loaded into the system memory 13***. The dispatch manager 17 executes the list of tasks contained therein to cause all required BIOS modules to be ***loaded into the system memory 13*** and must be executed. (Emphasis added)

Furthermore, as recited in column 3, lines 1-7,

After the computer is turned on, the minimal initialization code is executed to initialize the CPU ***and the system memory 13***. The dispatch manager is copied from the critical nonvolatile storage device to the ***system memory***. The dispatch manager ***sequentially executes the predetermined number of tasks to initialize the computer***.

It is clear that the dispatch operations are performed after system memory has been initialized and is thus the system memory is not only available, it is also employed under *Stevens*. Accordingly, *Stevens* does not teach the element of executing the firmware modules in a pre-memory execution environment prior to the initialization and availability of system memory. Furthermore, such an element is not suggested by either the *Stevens* or *ReStevens* reference, nor would one of ordinary skill in the art be motivated to do so. The availability of system memory makes the dispatch operations

much simpler than without system memory, since various data may simply be stored in any manner in the system memory, separate data structures may be used to store temporal data relating to the dispatch, *etc.*

Conclusion

Overall, none of the references singly or in any motivated combination disclose, teach, or suggest what is recited in the independent claims. Thus, independent claims 1, 15, 25, and 34 are now in condition for allowance. The dependent claims that depend directly or indirectly on these independent claims are likewise allowable based on at least the same reasons and based on the recitations contained in each dependent claim.

If the undersigned attorney has overlooked a teaching in any of the cited references that is relevant to the allowability of the claims, the Examiner is requested to specifically point out where such teaching may be found. Further, if there are any informalities or questions that can be addressed via telephone, the Examiner is encouraged to contact the undersigned attorney at (206) 292-8600.

Charge Deposit Account

Please charge our Deposit Account No. 02-2666 for any additional fee(s) that may be due in this matter, and please credit the same deposit account for any overpayment.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: Dec. 13, 2005



Anthony H. Azure
Reg. No. 52,580

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1030